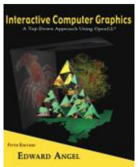


Name: ______________________ **Term # ____**

Homework 12 SOLUTIONS

(400 points)

NOTE: **Chapter 12 of the textbook shows the curves and surfaces.**
Part A is intended to be done by hand.
Part B is an OpenGL application.

_A._ (200 pts) Paper and Pencil

(Guidelines: *Read the material from the textbook chapter, you can use textbook figures to exemplify your answer, use keywords, summarize your answer, but the answer* ***cannot be longer the 7 lines!****)*

(There are 27 questions each worth 5 to 10 points)

12.1 REPRESENTATION OF CURVES AND SURFACES

a. Explain.
ANSWER:

12.1.1 Explicit Representation

a. Write it for curves, lines, circles, and surfaces.
ANSWER

12.1.2 Implicit Representation

a. Write it for curves, lines, circles, and surfaces.
ANSWER

12.1.3 Parametric Representation

a. Write it for curves, lines, circles, and surfaces.
ANSWER

12.1.4 Parametric Polynomial Curves Representation

a. Write it for curves.
ANSWER

12.1.5 Parametric Polynomial Surfaces Representation

a. Write it for surfaces.
ANSWER

12.2 DESIGN CRITERIA

a. Explain.
ANSWER:

12.3 PARAMETRIC CUBIC POLYNOMIAL CURVES

a. Write p(u)..
ANSWER:

12.4 INTERPOLATION

a. Write interpolation matrix.
ANSWER:

12.4.1 Blending Functions

a. Write them and draw them.
ANSWER

12.4.2 The Cubic Interpolating Patch

a. Write p(u,v).
ANSWER

12.5 HERMITE CURVES AND SURFACES

a. Explain.
ANSWER

12.5.1 The Hermite form

a. Write p(u)" and the Hermite Geometry Matrix.
ANSWER

12.5.2 Geometric and Parametric Continuity

a. Explain.
ANSWER

12.6 BEZIER CURVES AND SURFACES

a. Write the Bezier Geometry Matrix and Blending Functions.
ANSWER:

12.6.1 BEZIER Surface Patches

a. Write p(u,v).
ANSWER

12.7 CUBIC B-SPLINES

Explain
ANSWER:

12.7.1 The Cubic B-Spline Curve

What are the B-spline geometry matrix and blending functions.
ANSWER:

12.7.2 B-Splines and Basis

Explain
ANSWER:

12.7.3 Spline Surfaces

a. Write p(u,v).
ANSWER:

12.8 GENERAL B-SPLINES

a. Write p(u).
ANSWER:

12.8.4 NURBS
a. Write p(u).
ANSWER:

12.9 RENDERING CURVES AND SURACES
a. Explain.
ANSWER:

12.9.1 Polynomial Evaluation Methods
a. Explain.
ANSWER:

12.9.2 Recursive Subdivision of Bezier Polynomials
a. Write the values of the resulting points. Take a numerical example you will be tested on this.
ANSWER:

12.9.3 Rendering Other Polynomial Curves by Subdivision
a. Write the conversion matrices for the Interpolation and B-Splines to Bezier.
ANSWER:

12.9.4 Subdivision of Bezier Surfaces
a. Explain.
ANSWER:

12.11 ALGEBRAIC SURFACES
a. Explain.
ANSWER:

12.11.1 Quadrics
a. Write q(x,y,z).
ANSWER:

12.11.2 Rendering of Surfaces by Ray Casting
a. Write n and p(a)
ANSWER:

12.11.3 Subdivision Curves and Surfaces
a. Write s0 to s6
ANSWER:

12.11.4 Mesh Subdivision
a. Explain and Write p
ANSWER:

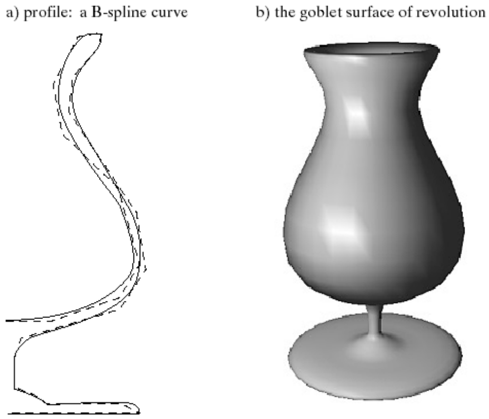
B. (200 pts) Visual Studio 2008 C++ Project Bezier Surfaces of Revolution

Create Visual Studio 2008 C++, Empty Project, Homework12:

It is often convenient to express the profile using Bezier or B-spline curves. We do this by selecting L+1 control points (Xk, Zk) and using them to create the curve

$$(X(v), Z(v)) = \sum_{k=0}^{L} (X_k, Z_k) N_{k,m}(v)$$

Below is an example of using a profile as a B-spline and it's surface of revolution.



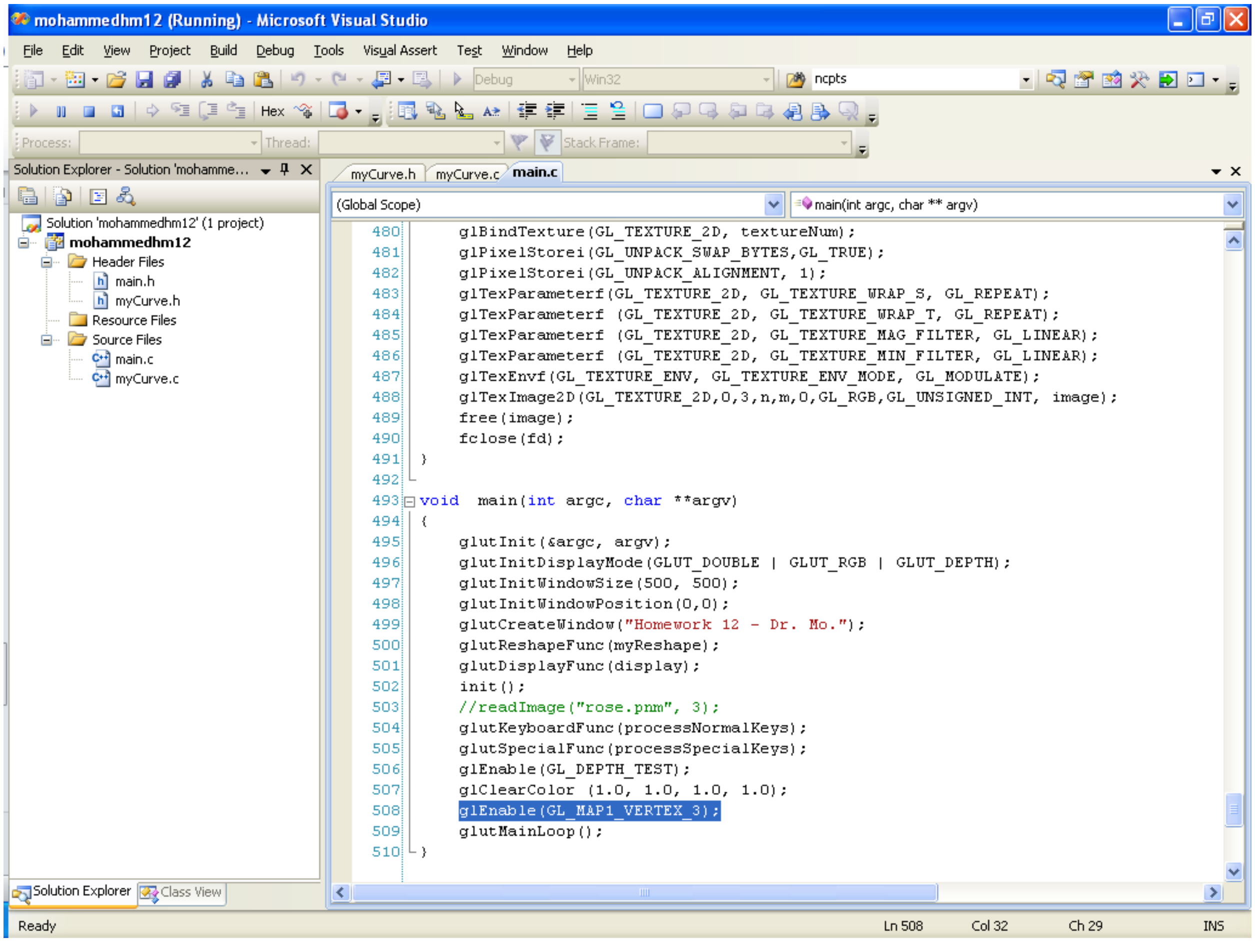
Given below is the 10 points of a Bezier mystery profile.

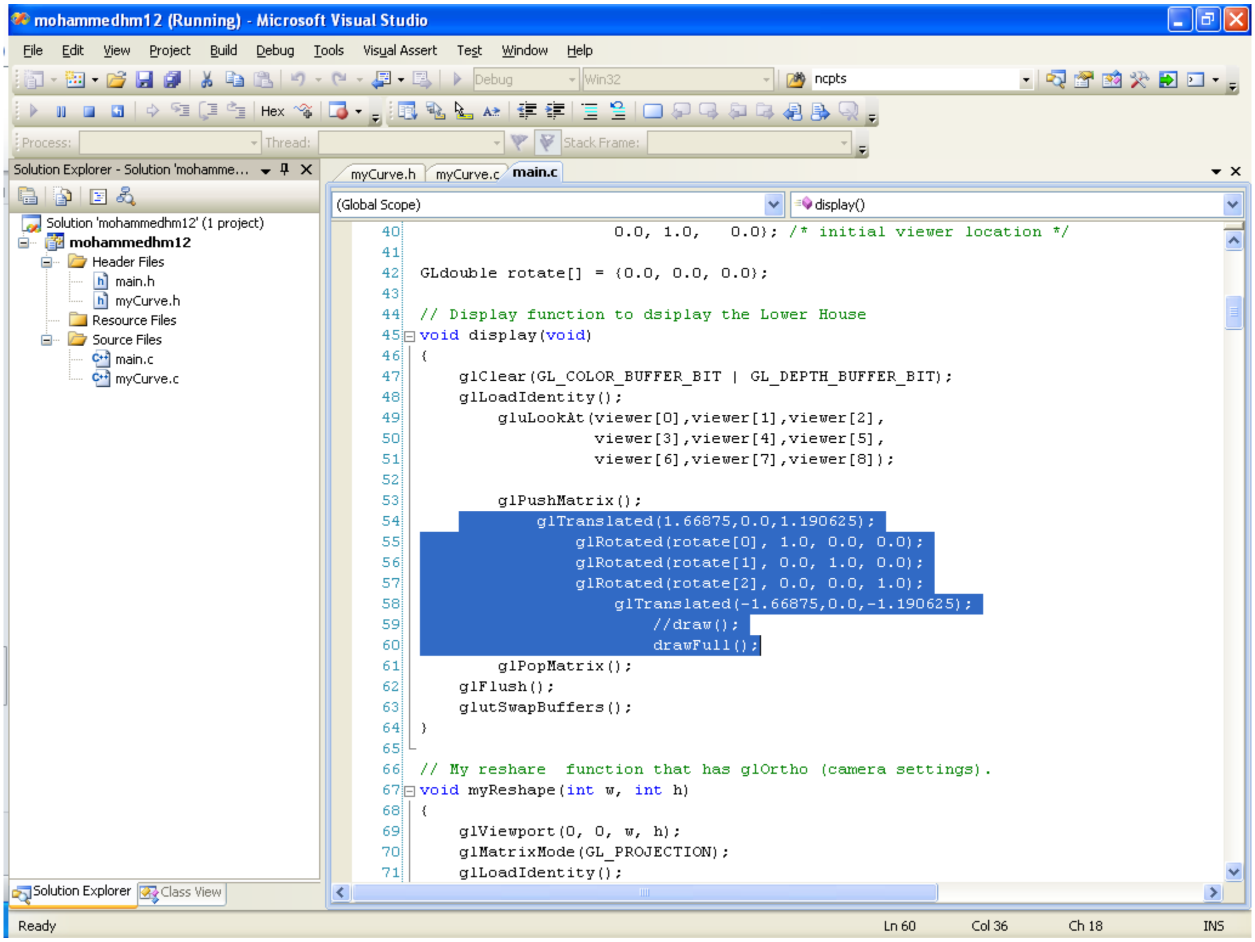
I	X	Z
0	1.4	2.25
1	1.3375	2.38125
2	1.4375	2.38125
3	1.5	2.25
4	1.75	1.725
5	2	1.2
6	2	0.75
7	2	0.3
8	1.5	0.075
9	1.5	0

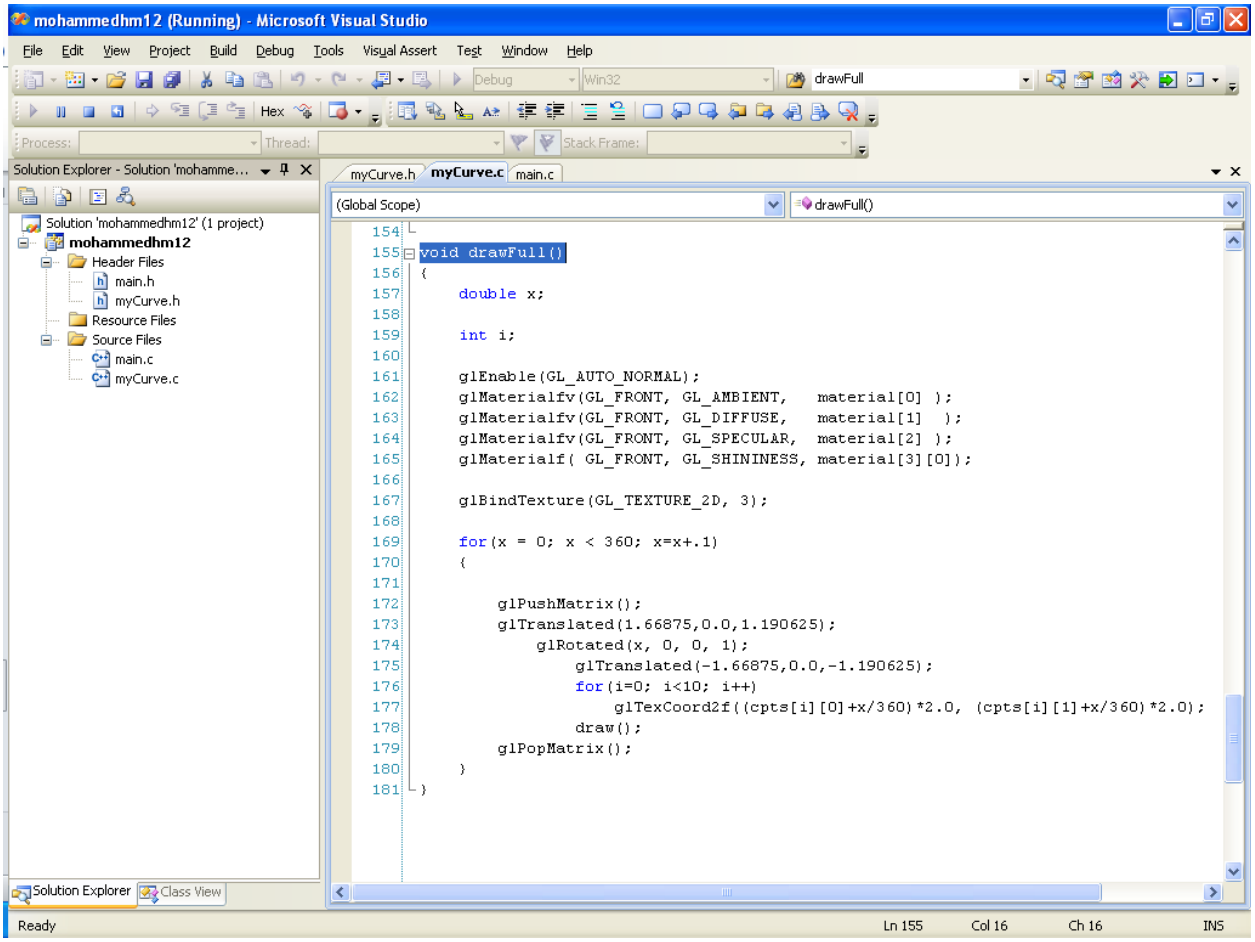
Render the a) and b) as in the figure above for these 10 points.

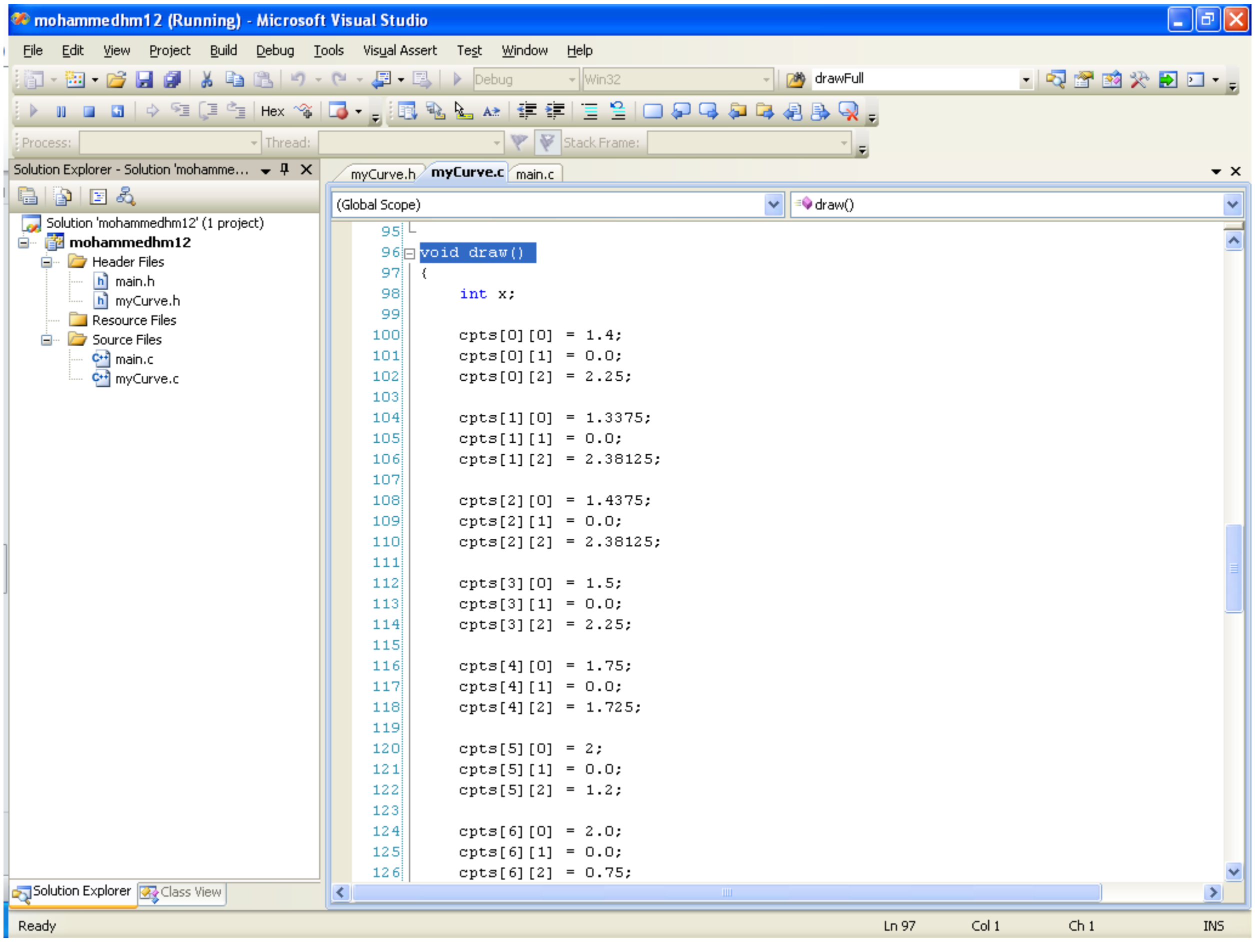
Build and run this Project: Insert a screenshot of your output.

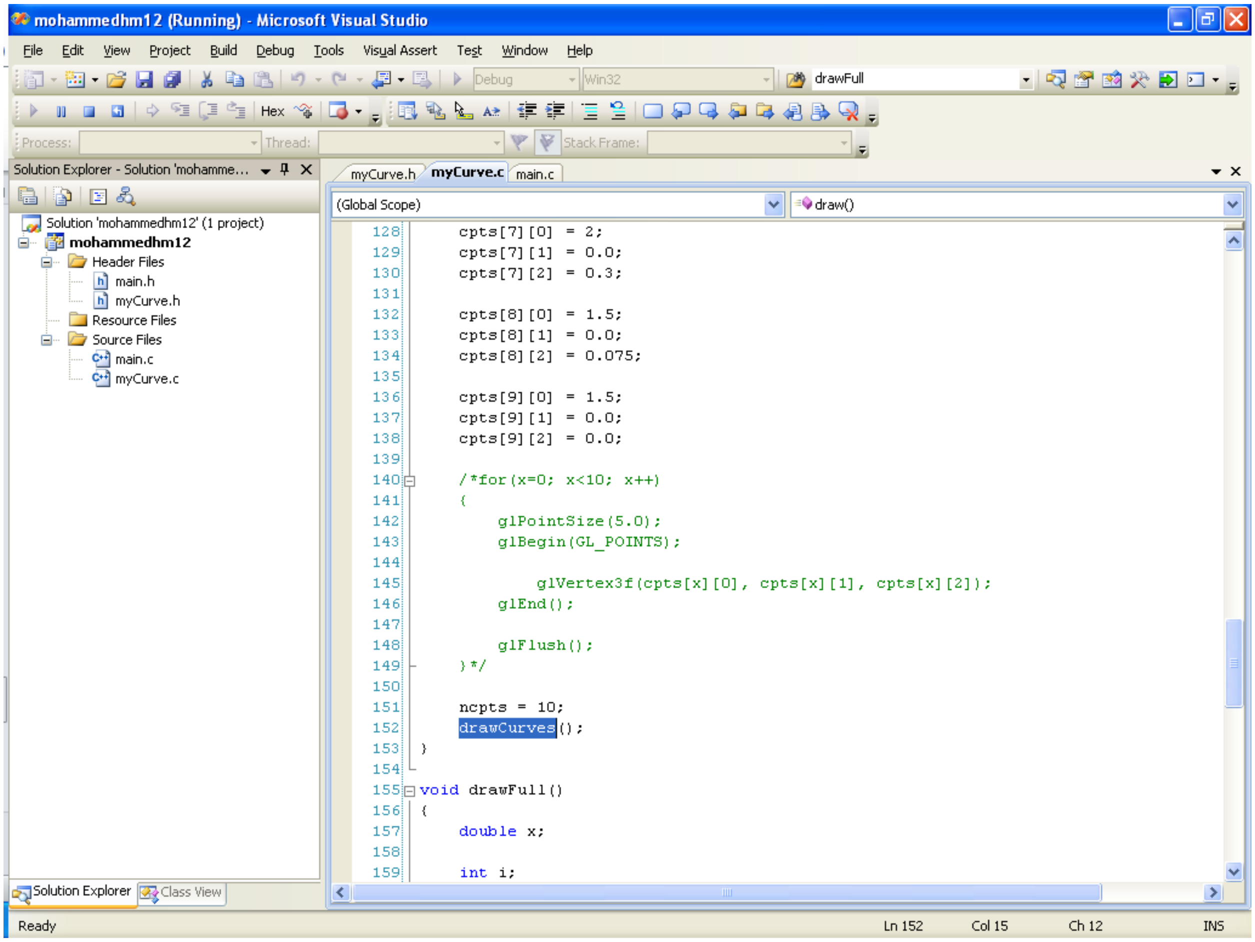
ANSWER:

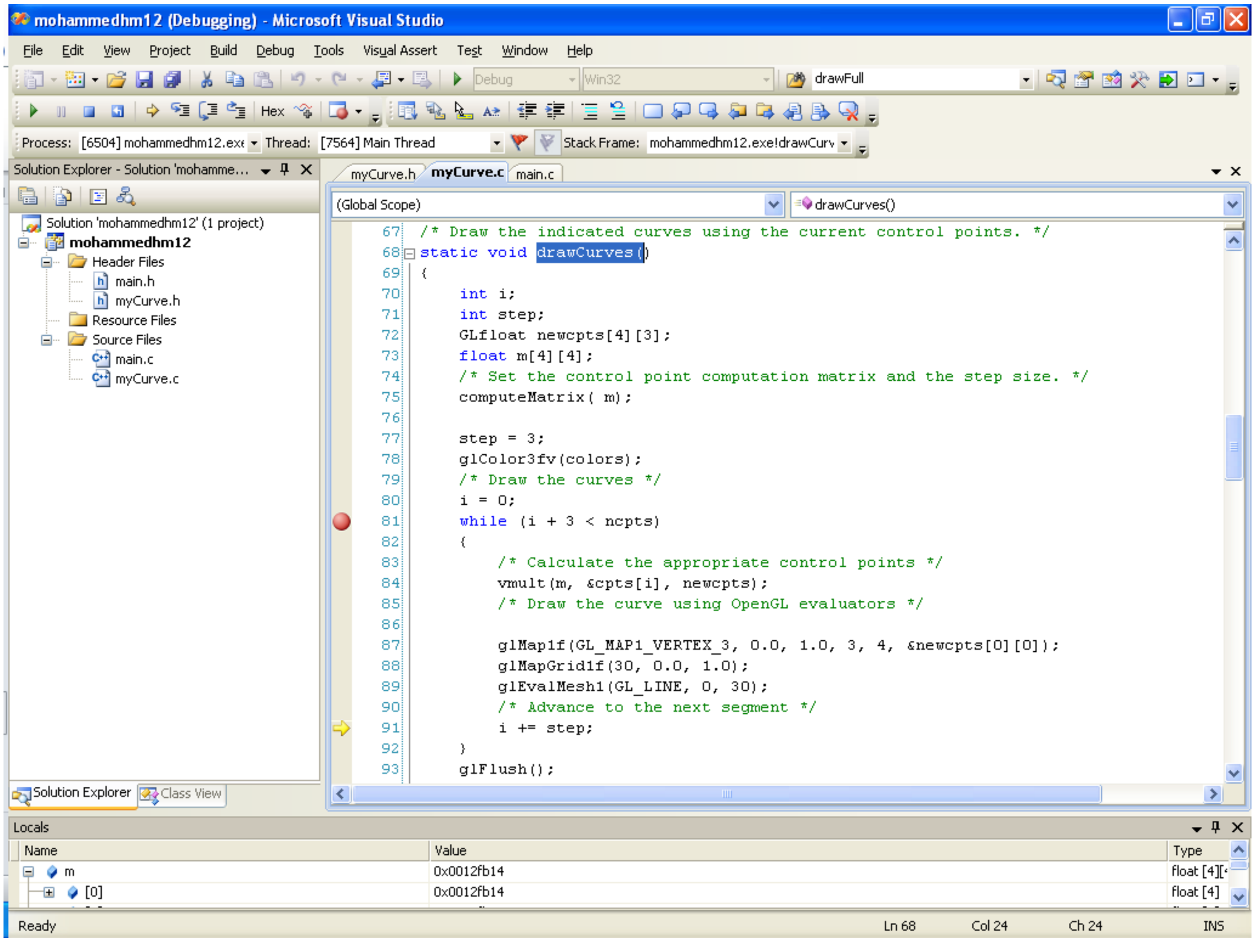

mohammedhm12 (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Tools Visual Assert Test Window Help
Solution Explorer - Solution 'mohamme...
Solution 'mohammedhm12' (1 project)
mohammedhm12
Header Files
main.h
myCurve.h
Resource Files
Source Files
main.c
myCurve.c
myCurve.h
myCurve.c
main.c
(Global Scope)
main(int argc, char ** argv)
glBindTexture(GL_TEXTURE_2D, textureNum);
glPixelStorei(GL_UNPACK_SWAP_BYTES,GL_TRUE);
glPixelStorei(GL_UNPACK_ALIGNMENT, 1);
glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
glTexParameterf (GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);
glTexParameterf (GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
glTexParameterf (GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
glTexImage2D(GL_TEXTURE_2D,0,3,n,m,0,GL_RGB,GL_UNSIGNED_INT, image);
free(image);
fclose(fd);
}
void main(int argc, char **argv)
{
glutInit(&argc, argv);
glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
glutInitWindowSize(500, 500);
glutInitWindowPosition(0,0);
glutCreateWindow("Homework 12 - Dr. Mo.");
glutReshapeFunc(myReshape);
glutDisplayFunc(display);
init();
//readImage("rose.pnm", 3);
glutKeyboardFunc(processNormalKeys);
glutSpecialFunc(processSpecialKeys);
glEnable(GL_DEPTH_TEST);
glClearColor (1.0, 1.0, 1.0, 1.0);
glEnable(GL_MAP1_VERTEX_3);
glutMainLoop();
}
Solution Explorer
Class View
Ready
Ln 508
Col 32
Ch 29
INS


mohammedhm12 (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Tools Visual Assert Test Window Help
Solution Explorer - Solution 'mohamme...
Solution 'mohammedhm12' (1 project)
mohammedhm12
Header Files
main.h
myCurve.h
Resource Files
Source Files
main.c
myCurve.c
myCurve.h
myCurve.c
main.c
(Global Scope)
display()
0.0, 1.0, 0.0}; /* initial viewer location */
GLdouble rotate[] = {0.0, 0.0, 0.0};
// Display function to dsiplay the Lower House
void display(void)
{
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
glLoadIdentity();
gluLookAt(viewer[0],viewer[1],viewer[2],
viewer[3],viewer[4],viewer[5],
viewer[6],viewer[7],viewer[8]);
glPushMatrix();
glTranslated(1.66875,0.0,1.190625);
glRotated(rotate[0], 1.0, 0.0, 0.0);
glRotated(rotate[1], 0.0, 1.0, 0.0);
glRotated(rotate[2], 0.0, 0.0, 1.0);
glTranslated(-1.66875,0.0,-1.190625);
//draw();
drawFull();
glPopMatrix();
glFlush();
glutSwapBuffers();
}
// My reshape function that has glOrtho (camera settings).
void myReshape(int w, int h)
{
glViewport(0, 0, w, h);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
Solution Explorer
Class View
Ready
Ln 60
Col 36
Ch 18
INS


mohammedhm12 (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Tools Visual Assert Test Window Help
Debug
Win32
drawFull
Hex
Process:
Thread:
Stack Frame:
Solution Explorer - Solution 'mohamme...
Solution 'mohammedhm12' (1 project)
mohammedhm12
Header Files
main.h
myCurve.h
Resource Files
Source Files
main.c
myCurve.c
myCurve.h
myCurve.c
main.c
(Global Scope)
drawFull()
void drawFull()
{
double x;
int i;
glEnable(GL_AUTO_NORMAL);
glMaterialfv(GL_FRONT, GL_AMBIENT, material[0]);
glMaterialfv(GL_FRONT, GL_DIFFUSE, material[1]);
glMaterialfv(GL_FRONT, GL_SPECULAR, material[2]);
glMaterialf(GL_FRONT, GL_SHININESS, material[3][0]);
glBindTexture(GL_TEXTURE_2D, 3);
for(x = 0; x < 360; x=x+.1)
{
glPushMatrix();
glTranslated(1.66875,0.0,1.190625);
glRotated(x, 0, 0, 1);
glTranslated(-1.66875,0.0,-1.190625);
for(i=0; i<10; i++)
glTexCoord2f((cpts[i][0]+x/360)*2.0, (cpts[i][1]+x/360)*2.0);
draw();
glPopMatrix();
}
}
Solution Explorer
Class View
Ready
Ln 155
Col 16
Ch 16
INS


mohammedhm12 (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Tools Visual Assert Test Window Help
Solution Explorer - Solution 'mohamme...
Solution 'mohammedhm12' (1 project)
mohammedhm12
Header Files
main.h
myCurve.h
Resource Files
Source Files
main.c
myCurve.c
myCurve.h
myCurve.c
main.c
(Global Scope)
draw()
void draw()
{
int x;
cpts[0][0] = 1.4;
cpts[0][1] = 0.0;
cpts[0][2] = 2.25;
cpts[1][0] = 1.3375;
cpts[1][1] = 0.0;
cpts[1][2] = 2.38125;
cpts[2][0] = 1.4375;
cpts[2][1] = 0.0;
cpts[2][2] = 2.38125;
cpts[3][0] = 1.5;
cpts[3][1] = 0.0;
cpts[3][2] = 2.25;
cpts[4][0] = 1.75;
cpts[4][1] = 0.0;
cpts[4][2] = 1.725;
cpts[5][0] = 2;
cpts[5][1] = 0.0;
cpts[5][2] = 1.2;
cpts[6][0] = 2.0;
cpts[6][1] = 0.0;
cpts[6][2] = 0.75;
Solution Explorer
Class View
Ready
Ln 97
Col 1
Ch 1
INS


mohammedhm12 (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Tools Visual Assert Test Window Help
Debug
Win32
drawFull
Hex
Process:
Thread:
Stack Frame:
Solution Explorer - Solution 'mohamme...
Solution 'mohammedhm12' (1 project)
mohammedhm12
Header Files
main.h
myCurve.h
Resource Files
Source Files
main.c
myCurve.c
Solution Explorer
Class View
myCurve.h
myCurve.c
main.c
(Global Scope)
draw()
cpts[7][0] = 2;
cpts[7][1] = 0.0;
cpts[7][2] = 0.3;
cpts[8][0] = 1.5;
cpts[8][1] = 0.0;
cpts[8][2] = 0.075;
cpts[9][0] = 1.5;
cpts[9][1] = 0.0;
cpts[9][2] = 0.0;
/*for(x=0; x<10; x++)
{
glPointSize(5.0);
glBegin(GL_POINTS);
glVertex3f(cpts[x][0], cpts[x][1], cpts[x][2]);
glEnd();
glFlush();
}*/
ncpts = 10;
drawCurves();
}
void drawFull()
{
double x;
int i;
Ready
Ln 152
Col 15
Ch 12
INS



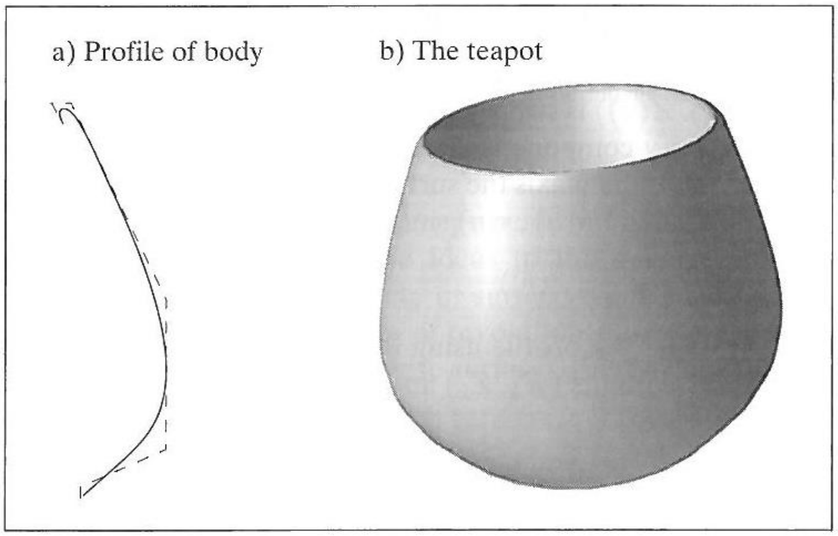

```
/* Draw the indicated curves using the current control points. */
static void drawCurves()
{
    int i;
    int step;
    GLfloat newcpts[4][3];
    float m[4][4];
    /* Set the control point computation matrix and the step size. */
    computeMatrix( m);

      step = 3;
    glColor3fv(colors);
    /* Draw the curves */
    i = 0;
    while (i + 3 < ncpts)
    {
        /* Calculate the appropriate control points */
        vmult(m, &cpts[i], newcpts);
        /* Draw the curve using OpenGL evaluators */

        glMap1f(GL_MAP1_VERTEX_3, 0.0, 1.0, 3, 4, &newcpts[0][0]);
        glMapGrid1f(30, 0.0, 1.0);
        glEvalMesh1(GL_LINE, 0, 30);
        /* Advance to the next segment */
        i += step;
```

```
    }
    glFlush();
}
```

FIGURE 10.55 Bezier-based profiles for the teapot body.



condition insures that the different Bezier curves blend together with a continuous derivative (recall Section 10.1.2).

The lid of the teapot is also a surface of revolution, and is described in Case Study 10.8.